

Tartu Ülikool  
Loodus- ja täppisteaduste valdkond  
Tehnoloogiainstituut

Jüri Gramann

# **Eeltöö videohõiveks Genesys 2 arendusplaadil**

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendaja:  
MSc Margus Rosin

Tartu 2017

# Resümee/Abstract

## Eeltöö videohõiveks Genesys 2 arendusplaadil

Videomeedium on kiiresti kasvav ja arenev valdkond. Videomeediumi tootmise üks suur osa on videohõive. Bakalaureusetöö eesmärgiks on teha eeltööd videohõiveks Digilent, Inc. toodetud Genesys 2 arendusplaadil. Kasutatavaks videoliideseks on HDMI. Töö praktilises osas hõivati arendusplaadiga ühe kaadri jagu andmeid, saadeti see arvutisse ja salvestati BMP pildifailina. Lisaks realiseeriti EDID, et arvuti arendusplaati kuvarina tuvastaks.

Tulemusteks oli arendusplaadi tuvastamine kuvarina ja ühe kaadri jagu info salvestamine. Salvestud kaader ei olnud perfektne, sest pilt oli nihkes. Lisaks leiti, et kirjutatud loogika kasutas vähe FPGA ressursse. See tähendab, et kasutatavat loogikat saaks realiseerida ka väiksemal FPGA-l või lisada kasutatavale loogikale täiendavaid funktsioone. Antud töös tehti ka järeldusi, mida oleks vaja lisada, et realiseerida kasutatud arendusplaadil täielik videohõive.

**CERCS:** T121 Signaalitöötlus; T180 Telekommunikatsioonitehnoloogia

**Märksõnad:** FPGA, HDMI, videohõive, Genesys 2, VHDL

## Preparation for video capture on Genesys 2 development board

Video media content is fast-growing area of media. Big part of creating video is video capturing. The purpose of this bachelor thesis is to prepare for video capturing on Digilent, Inc. Genesys 2 development board using HDMI. In practice the development board was used to capture one frame, send it to a computer and save it as a BMP image. In addition, an EDID was implemented to recognize the development board as a display.

As a result, a computer recognized the development board as a display and one frame was successfully transmitted and saved to the computer. The saved frame was not perfect: it had a small shift in it. In addition, the implemented logic used very little FPGA resources. That means the logic could be implemented on a smaller FPGA or there could be space for additional functions. Also in this thesis, a conclusion was made about what needs to be added to implement full video capturing functionality.

**CERCS:** T121 Signal processing; T180 Telecommunication engineering

**Keywords:** FPGA, HDMI, video capture, Genesys 2, VHDL

# Sisukord

Jooniste loetelu .....	6
Tabelite loetelu .....	7
Lühendid, mõisted .....	8
1 Sissejuhatus .....	10
1.1 Probleemi tutvustus .....	10
1.2 Töö eesmärk ja ülevaade .....	10
2 Ülevaade probleemist .....	11
2.1 Videohõive .....	11
2.2 HDMI .....	12
2.3 TMDS .....	13
2.4 FPGA .....	14
3 Metoodika .....	16
3.1 Kasutatavad tööriistad .....	16
3.1.1 Genesys 2 .....	16
3.1.2 VHDL ja Vivado .....	17
3.2 EDID .....	18
3.3 Andmevoo töötlus .....	19
3.3.1 Paralleliseerimine .....	19
3.3.2 Dekodeerimine .....	20
3.3.3 Sünkronisatsioon .....	21
3.4 Andmete salvestamine .....	21
3.5 Andmevahetus .....	22
3.6 Taktsignaalide haldus .....	23
3.6.1 Taktsignaalide marsruutimine .....	23
3.6.2 Taktsignaalide genereerimine .....	24

3.7	Tööga kaasas olevad failid .....	24
4	Tulemused ja järeldused .....	25
4.1	Taktsignaalid .....	25
4.2	Kasutatud ressursid .....	25
4.3	Seadme tuvastamine .....	26
4.4	Kaadrihõive .....	27
5	Kokkuvõte .....	31
	Kasutatud kirjandus .....	32
	Lisad .....	36
	Lisa 1 TMDS kodeerija loogika [5].....	36
	Lisa 2 EDID väärtuste tabel.....	37
	Lisa 3 TMDS dekodeeri loogika [5].....	39
	Lisa 4 BMP faili päis .....	40

# Jooniste loetelu

Joonis 1. FPGA baasil videohõive plokk skeem [18].....	11
Joonis 2. A tüüpi HDMI kontaktid [9] .....	12
Joonis 3. TMDS arhitektuur [5].....	14
Joonis 4. FPGA sisemine ehitus .....	15
Joonis 5. Süsteemi loogika .....	16
Joonis 6. Genesys 2 arendusplaat .....	17
Joonis 7. EDID struktuur [7] .....	19
Joonis 8. SerDesi põhimõte .....	20
Joonis 9. Taktsignaali võrk .....	25
Joonis 10. Arendusplaadi tuvastamine arvutis .....	26
Joonis 11. Väljavõtte terminaliaknast pärast kaadri hõivamist .....	27
Joonis 12. Esimene arusaadav kaader.....	27
Joonis 13. Hõivatud kaader eesti lipust .....	28
Joonis 14. Hõivatud kaader Tehnoloogiainstituudi kodulehest.....	29
Joonis 15. Algne testmuster [33] .....	29
Joonis 16. Arendusplaadiga salvestatud testmuster.....	30
Joonis 17. Hõivatud testmusteri vead suurendatult .....	30

# Tabelite loetelu

Tabel 1. FPGA ressursside kasutamine .....	26
--	----

# Lühendid, mõisted

4K – viitab resolutsioonile 4096 x 2160. Vahetevahel viitab see ka resolutsioonile 3840 x 2160, aga selle õige nimi on 4K Ultra HD. [1]

8K – teise nimega 8K Ultra HD viitab resolutsioonile 7680 x 4320. [1]

ASIC (*Application-Specific Integrated Circuit*) – integraalskeem, mis on disainitud kindlaks rakenduseks. [2]

BMP (*Bitmap*) – rastergraafika failitüüp. [3]

CMT (*Clock Management Tile*) – taktihaldus ala, mis on ehitatud Xilinx, Inc. seitsmenda seeria FPGA-desse. Üks CMT sisaldab endas MMCM-i ja PLL-i. [4]

DDC (*Display Data Channel*) – kuvari andmete kanal, mis kasutab suhtluseks I2C standardit. [5]

DDR (*Double Data Rate*) – meetod sünkroonseks andmevahetuseks, kus väärust loetakse nii tõusval kui langeval taktsignaali.

DDR3 SDRAM – kiire muutmälu tüüp.

DPTI (*Digilent Parallel Transfer Interface*) – Digilendi paralleel andmevahetuse liides nende toodetud arendusplaatide ja personaalarvutite vahel. [6]

DVI (*Digital Video Interface*) – digitaalvideoliides kuvari ja personaalarvuti ühendamiseks, mis kinnitati 1999. aastal. [5]

EDID (*Extended Display Identification Data*) – täiendatud kuvari identifitseerimise andmed, mida saadetakse üle DDC signaaliradade ja mida kasutatakse kuvari tuvastamiseks. [7]

FIFO (*First in, first out*) – järjenditüüp, kus esimesena sisestatud andmed väljastatakse esimesena ehk järjekord.

FPGA (*Field-programmable gate array*) – integraalskeem, mis koosneb konfigureeritavate loogikaplokkide maatriksist, konfigureeritavatest sisend-väljund viikudest ja nendevahelisest programmeeritavatest sisemistest vahe-ühendustest. [8]



HDMI (*High-Definition Multimedia Interface*) – kõrglahutusega multimeediumliides digitaalse video ja audiosignaali edastamiseks. [9]

I2C (*Inter-Integrated Circuit*) – 1980-datel Philipsi poolt välja töötatud suhtlusstandard ühel trükkplaadil olevate komponentide vahel. [10]

LUT (*LookUp Table*) – otsingutabel, mis paneb kindlatele sisenditele vastavusse kindlad väljundid.

MMCM (*Mixed-Mode Clock Manager*) – Xilinx, Inc. kasutatav PLL edasiarendus nende seitsmenda seeria FPGA-des. [4]

PCI (*Peripheral Component Interconnect*) – personaalarvuti siin välisseadmeühenduseks

PLL (*Phase-locked loop*) – elektroonikaskeem, mida kasutatakse taktsignaali genereerimiseks, taastamiseks ja stabiliseerimiseks. PLL kasutab toimimiseks muutliku võnkumisega ostsillaatorit. [11]

RAM (*Random-access memory*) – muutmälu

RGB (*Red Green Blue*) – värvimudel, millega saab kujutada kõiki värve. Kasutatavad värvid on R – punane, G – roheline ja B – sinine.

SerDes (*Serializer/Deserializer*) – jada-paralleelmuundur, mis muudab jadaandmed paralleelandmeteks või vastupidi. [12]

TMDS (*Transition-minimized differential signaling*) – madalapingeline diferentsiaalne signaaliedastusmeetod, mis dekodeerib saadetavad andmed, et tasakaalustada signaaliedastust ja vähendada elektromagnetilisi häireid. [5]

USB (*Universal Serial Bus*) – universaalne järjestiksiin arvuti välisseadmete ühendamiseks.

VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) – 1980-ndatel Ameerika Ühendriikide Kaitseministeeriumi poolt välja töötatud riistvarakirjelduskeel. [13]

# 1 Sissejuhatus

Videomeedia vaatamine on kiiresti kasvav ja arenev meedium [14]. Videojagamisplatvorm YouTube.com on külastatavuselt teine kõige populaarsema, Google.com, järel ja videostiimimisplatvorm Twitch.tv on järjestuses 50. (27. aprill 2017 seisuga) [15]. Videomeedia üha kasvava populaarsuse tulemusena kasvab ka uute videote ja videostiimide loojate arv. Video salvestamiseks või striimimiseks on vaja videovoog muuta arvutile arusaadavaks, olgu selle videovoo päritoluks videomäng või kaamera. Protsessi, mille käigus kodeeritakse videovoog arvutile loetavaks ja taasesitatavaks, nimetatakse videohõiveks.

## 1.1 Probleemi tutvustus

Videohõive on protsess, mille käigus võetakse digitaalne või analoogne videovoog ja salvestatakse andmekandjale [16]. Videohõivet kasutatakse näiteks erinevate videoülekannete tegemisel ja salvestamisel. Arvutitele on olemas tarkvaralised videohõive programmid, mis kasutavad andmetötluseks arvuti keskprotsessorit või graafikaprotsessorit. Arvuti protsessorite kasutamine andmetötluseks võib aga põhjustada arvuti, kui terviküsteemi, aeglust. Lisaks võimaldab tarkvaraline videohõive salvestada ainult arvuti poolt genereeritud videovoogu, sest arvutitel üldjuhul puudub videosisend. Antud piirangute vältimiseks kasutatakse eraldiseisvat videohõive seadet. Väga levinud on seadmed, mis kasutavad PCI või USB liidest.

## 1.2 Töö eesmärk ja ülevaade

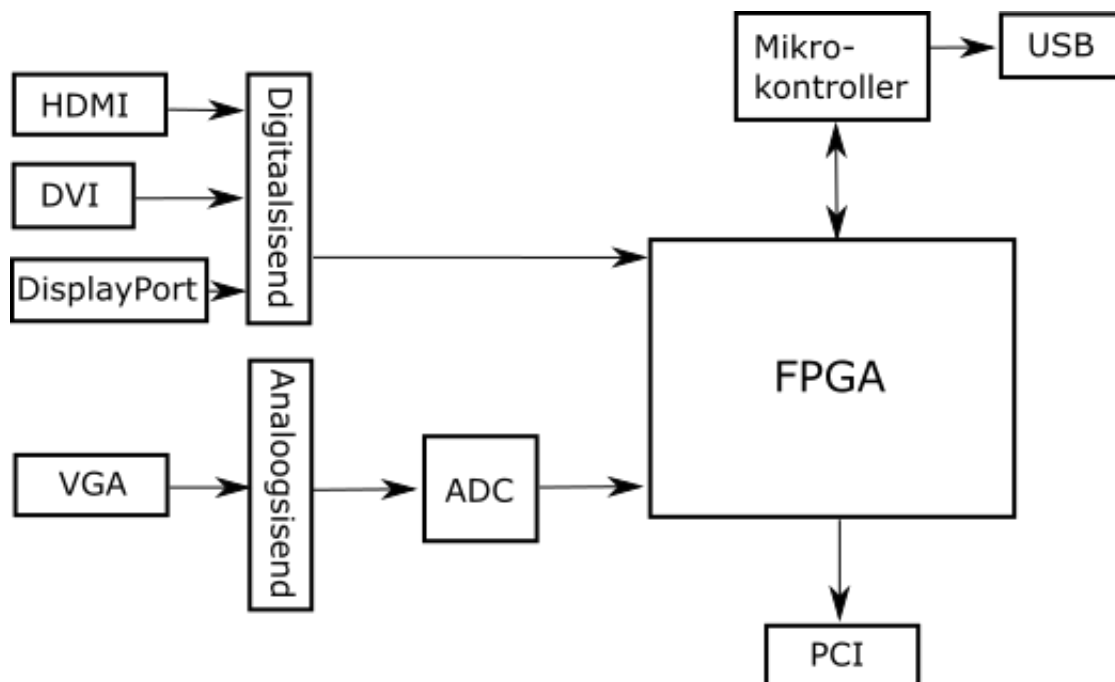
Bakalaureusetöö eesmärgiks on teha eeltööd videohõiveks Digilent, Inc. toodetud Genesys 2 arendusplaadil, kasutades HDMI liidest. Selleks tuleb tutvuda HDMI standardiga ja kirjutada tarkvara videosignaali vastuvõtmiseks kasutatavale arendusplaadile. Lisaks salvestada ühe kaadri jagu infot korraga ja saata see arendusplaadilt arvutisse kasutades USB-d.

Bakalaureusetöö sisuline osa jaotub neljaks struktureeritud peatükiks. Esimeses räägitakse digitaalse videohõivega kaasnevatest probleemidest. Teises peatükis pakutakse probleemidele üks lahendus ja kirjeldatakse antud probleemide lahenduskäiku. Kolmandas ja neljandas peatükis räägitakse saadud tulemustest ning tehakse nende põhjal järeldused.

## 2 Ülevaade probleemist

### 2.1 Videohõive

Tänapäeval on videovood suuremahulised ning kiiresti kasvavad, sest resolutsioonid ulatuvad 4K ja isegi 8K-ni. Näiteks väga levinud 1920x1080 resolutsioonil ning 60 Hz kaadrisagedusel on andmevahetuskiiirus  $1920 * 1080 * 24 \text{ biti} * 60 \text{ Hz} = 2,99 \text{ Gb/s}$  ja piksli taktsagedus on  $1920 * 1080 * 60 \text{ Hz} = 124,4 \text{ MHz}$ . Antud arvutused ei arvesta tühja alaga iga kaadri ümber ja kodeeringuga, mida tehakse näiteks HDMI signaaliga [9]. Klassikalised mikroprotsessorid ei ole piisavalt kiired selliste andmevoogudega töötamiseks. Selle tõttu kasutab näiteks Magewell Electronics Co. oma professionaalsetes videohõive toodetes Xilinx, inc. Artix-7 FPGA-d [17].

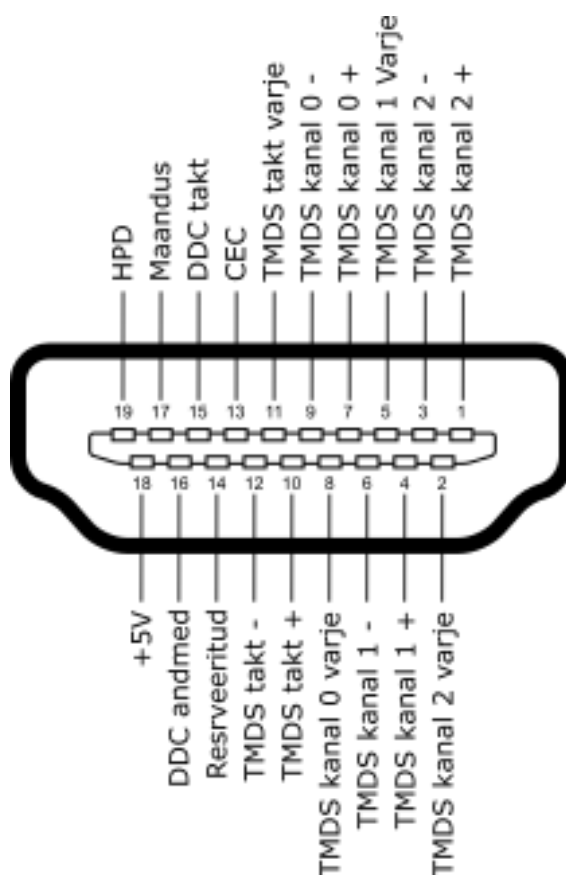


Joonis 1. FPGA baasil videohõive plokkskeem [18]

FPGA-ga videohõive seadme sisendiks on kas digitaalne või analoogne videosignaali. Kui tegemist on analoogsignaali on see vaja konverteriga digitaliseerida. Järgnevalt töötleb FPGA antud signaali vajalikku formaati ja sõltuvalt väljundist kas FPGA suhtleb ise üle PCI või saadab andmed täiendavale integraalskeemile (näiteks mikrokontroller), mis omakorda suhtleb üle USB. [18]

## 2.2 HDMI

HDMI ehk kõrglahutusega multimeediumiliides (*High-Definition Multimedia Interface*) on liides digitaalse videosignaali edastamiseks. Lisaks võimaldab see edastada audiosignaali. HDMI on DVI edasiarendus, mille versioon 1.0, mis võimaldab 1920 x 1080 resolutsioonil 60 Hz-st edastust, anti välja 2002. aastal. 2006. aastal väljastati versioon 1.3a, mis kahekordistas andmeedastuskiiruse. 2017. aasta alguses kuulutati välja versioon 2.1 [19]. Antud töös kasutatakse versiooni 1.3a, sest kasutataval arendusplaadil olev HDMI varjesti (*hider*) on selle standardiga ühilduv [20]. [9]



Joonis 2. A tüüpi HDMI kontaktid [9]

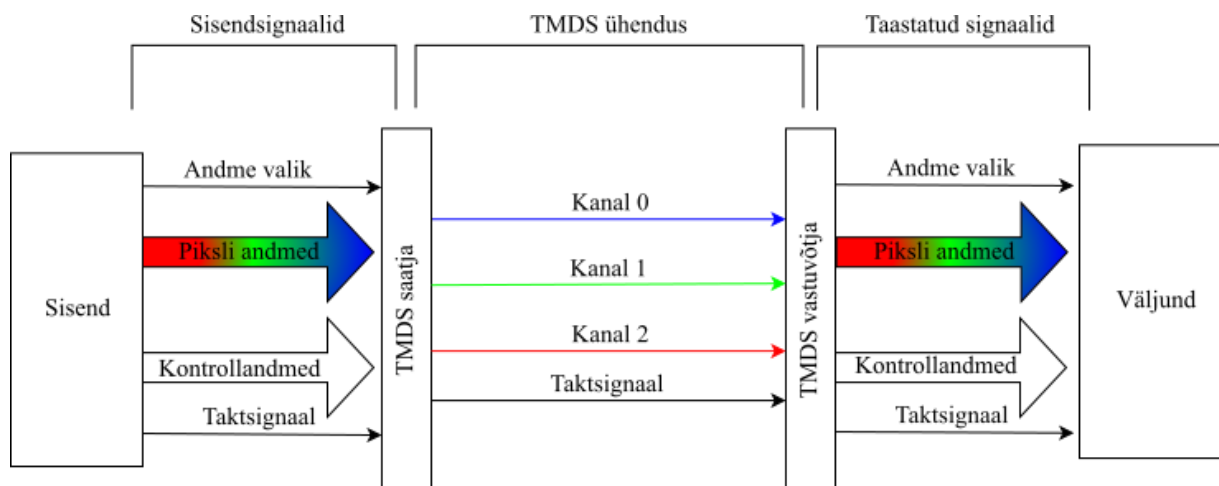
HDMI A tüüpi pistik kasutab 19 signaalirada, millest 12 on TMDS signaalidele. Iga TMDS signaal kasutab kahte signaalirada diferentsiaalsignaale ja ühte signaaliradade varjeks. Lisaks on kaks signaalirada DDC ühenduseks, millega vahetakse EDID (*Extended Display Identification Data*) andmeid, üks signaalirada koduelektroonika kontrolliks üle HDMI (CEC – *Consumer Electronics Control*), üks signaalirada käigultvahetuse tuvastamiseks (HPD - *Hot-Plug Detect*) ja üks +5V voolule. Üks andmerada on reserveeritud ja versioonil 1.3 seda ei kasutata. [9]

Üle DDC andmeradade saadetakse I2C standardit kasutades EDID infot. Baas EDID hoiab endas 128 baiti kuvari identifitseerimiseks vajalikku infot. Näiteks hoiab see endas monitori tootja infot, EDID versiooni numbrit, ekraani mõõtmeid ja kasutatavaid resolutsioone. Laiendatud EDID (E-EDID) suudab hoida endas lisaks 128 baasbaidile veel kuni 255 lisaplokki, millest igaüks on 128 baiti suur ja kus hoitakse täiendavat infot. [7]

HDMI saadetavad andmed jagunevad kolmeks perioodiks: videoandmed, andmesaar (*data island*), kontroll. Videoandmete perioodi ajal saadetakse piksli väärtuste infot. Andmesaare perioodil saadetakse audio või muud abiinfot pakettidena. Kontrollperioodil saadetakse kontrollandmeid, milleks on horisontaal- ja vertikaalsünkronisatsiooni signaalid ning erinevate andmesaarte alguste signaalid. Lisaks kasutatakse kontrollperioodi ka videoandmete ja andmesaarte perioodide eristamiseks üksteisest. Kontrollperiood peab jääma kahe ülejäänud perioodi vahele. [9]

## 2.3 TMDS

HDMI kasutab signaalide edastamiseks minimaalse üleminekuprotsessiga diferentsiaalset signaaliedastust ehk TMDS-i (*Transition Minimized Differential Signalling*). TMDS on madalapingeline diferentsiaalne signaaliedastusmeetod, mille eesmärk on tasakaalustada signaaliedastust, et vähendada elektromagnetilisi häireid ja võimaldada sellega kiiremat andmeedastust. HDMI kasutab korraga kolme TMDS kanalit, mis kodeerib ühe taktsignaali kohta kas piksli, andmesaare või kontrollandmeid, sõltuvalt perioodist. Kanal 0 kasutatakse üldjuhul sinise, kanal 1 rohelise ja kanal 2 punase värvi edastamiseks. Horisontaal- ja vertikaalsünkronisatsiooni kontrollsignaale saadetakse kanal 0-ga [5]. Kanal 1 ja kanal 2 saadetavaid kontrollsignaale kasutatakse erinevate andmesaarte alguste märkimiseks. [9]

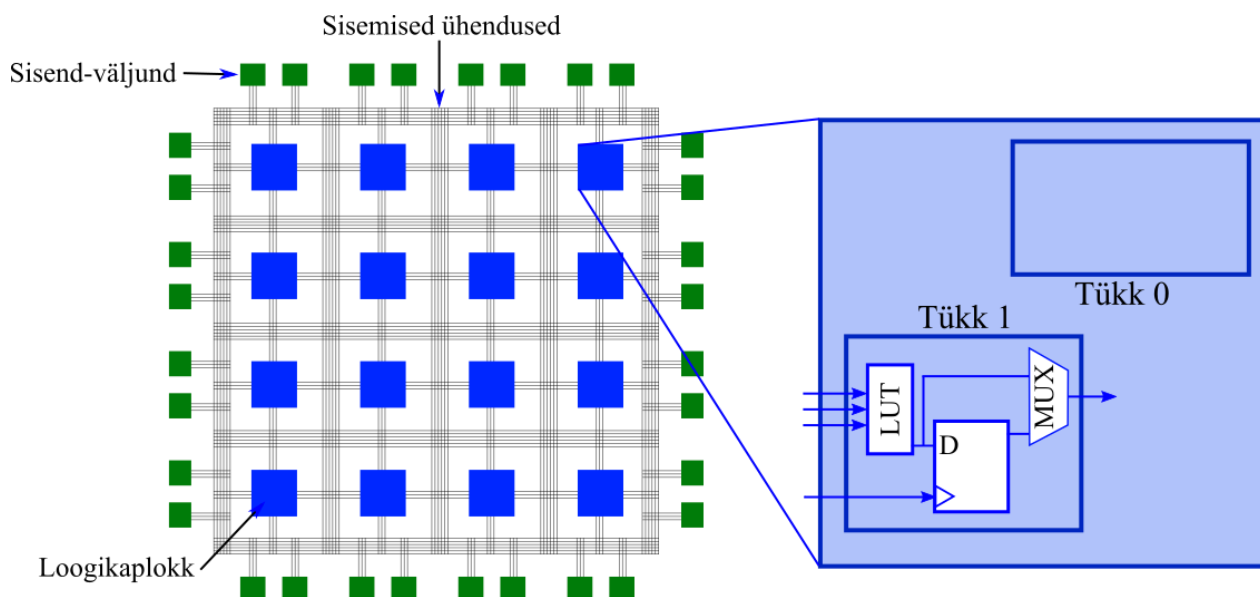


Joonis 3. TMDS arhitektuur [5]

TMDS kodeerija võtab sisendiks perioodist sõltuvalt 8 bitti piksli andmeid, 4 bitti andmesaare andmeid või 2 bitti kontrollandmeid ja väljastab kodeeritud kujul 10-bitise sümboli. Kontrollandmete saatmiseks kasutatakse ühte neljast sümbolist, millest igaüks vastab ühele kontrollandmete bittide olekule. Piksli andmete puhul töötab TMDS kodeerija kahes astmes. Esimeses astmes tekitatakse 9-bitine sümbol, kus biti väärtuste üleminekud on vähendatud. Teises astmes tekitatakse lõplik kümne bitine sümbol, mis on tasakaalustatud. Täpne algoritmi ülesehitus on saadaval lisas 1. Kodeerija väljastatud sümbolis on biti väärtuste muutumisi vähendatud, mille tulemusel on seda võimalik kiiremal sagedusel saata. [5]

## 2.4 FPGA

FPGA (*Field Programmable Gate Array*) on integraalskeem, mis koosneb konfigureeritavate loogikaplokkide maatriksist, konfigureeritavatest sisend-väljund viikudest ja nendevahelisest programmeeritavatest sisemistest vaheühendustest. Iga loogikaplokk koosneb tükkidest (*slice*), mis omakorda koosnevad LUT-idest, trigeritest ja multiplekseritest (MUX). LUT-e kasutatakse loogikaelementide asemel, sest neid on võimalik programmeerida, määrates kindlad väljundid kindlate sisendite kohta. Lisaks kiirendavad need andmetöötlust, sest andmed peavad läbima vajalikeks arvutusteks vähem loogikaelemente. [8] [21]



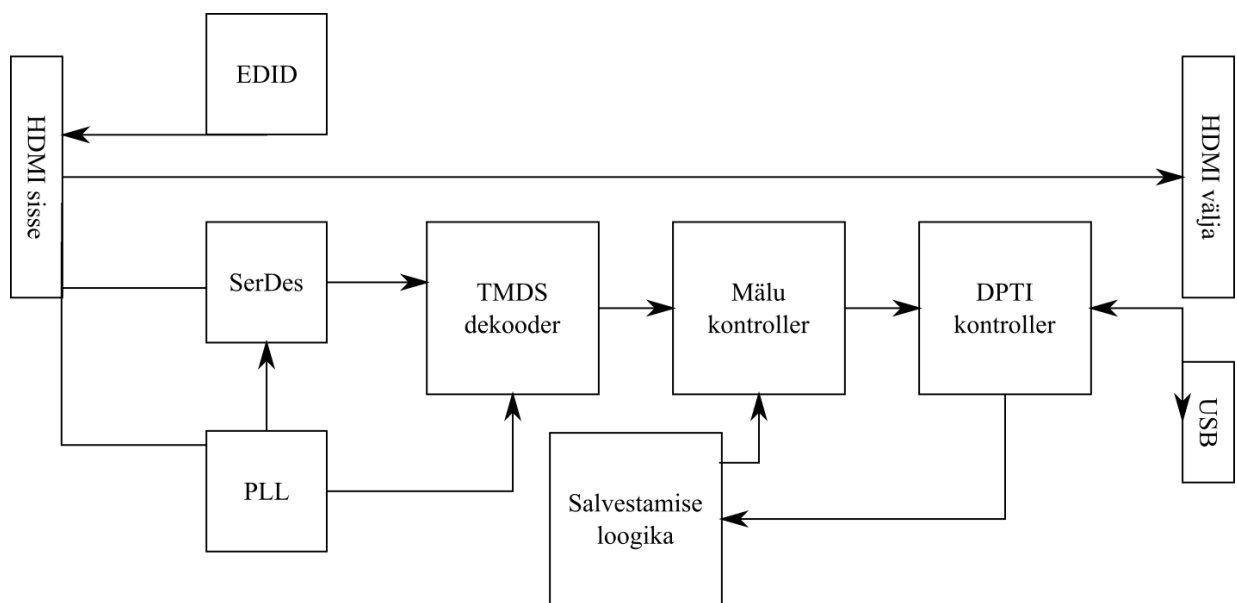
Joonis 4. FPGA sisemine ehitus

FPGA-des on andmete salvestamiseks sisemine mälu ja lisaks kasutakse tihti ka väliseid mäluseadmeid. Sisemine mälu jaguneb plokk-RAM-iks ja jagatud (*distributed*) RAM-iks. Plokk-RAM on eraldi element FPGA-s, mida saab kasutada ainult andmete hoiustamiseks. Jagatud RAM on aga tootjast sõltuvalt erinev. Intel FPGA (varasemalt Altera Corporation) seadmed kasutavad jagatud RAM-iks väiksemaid plokk-RAM-e. Xilinx, Inc. seadmed kasutavad aga loogikatükkide LUT-e andmete hoiustamiseks. Üldiselt eelistatakse plokk-RAM-i suuremate andmete hoiustamiseks ja jagatud RAM-i väiksemate andmete hoiustamiseks [22]. Sisemise RAM-i hulk on piiratud. Kui on vaja salvestada suuremaid andmeid, siis on vaja kasutada välist RAM-i. Väline mälu on küll sisemisest mälust aeglasem, aga selle pakutav mälumaht on kordades suurem. Näiteks on käesolevas töös kasutataval arendusplaadil 1 GB välist mälu ja 16 000 Kb plokk-RAM-i. [23]

FPGA alternatiiviks on ASIC (*Application Specific Integrated Circuits*). ASIC-u eelisteks on masstoodangu korral odavam hind, täielikult kohandatud funktsioonid ja väiksem suurus. FPGA-dega on lihtsam töötada, sest tarkvara teeb kasutaja jaoks kõik ära. Lisaks on alustamine odavam, sest ei pea hakkama kohe massiliselt tootma. Peamine eelis on aga võimalus FPGA-sid korduvalt ümberprogrammeerida. [24]

## 3 Metoodika

Videohõivekaardi töö saab jagada väiksemateks osadeks. Esimene osa paralleliseerib andmevoo ja dekodeerib info. Järgmine osa salvestab kaadri info ajutiselt, sest seadmest väljuv andmevahetuskirgus on piiratud. Selle probleemi lahendamiseks tuleb andmevoogu tihendada, vähendada video kaadrisagedust või teha mõlemat. Viimane osa suhtleb üle USB liidese arvutiga. Antud kontrolleri võtab vastu arvutist käsked ja saadab mälust töödeldud andmed arvutisse, kus need salvestatakse. Veel on vajalik realiseerida EDID kontrolleri, mis sisaldab EDID andmeid ja selle saatmise loogikat. EDID kontrolleri on vajalik, et videoallikas hakkaks üldse videosignaali saatma. Lisaks on HDMI sisend ka otse ühendatud HDMI väljundiga selleks, et vajadusel suunata salvestatav videosignaal monitorile. Kogu süsteem disainitakse VHDL-is ning implementeeritakse Genesys 2 arendusplaadil.



Joonis 5. Süsteemi loogika

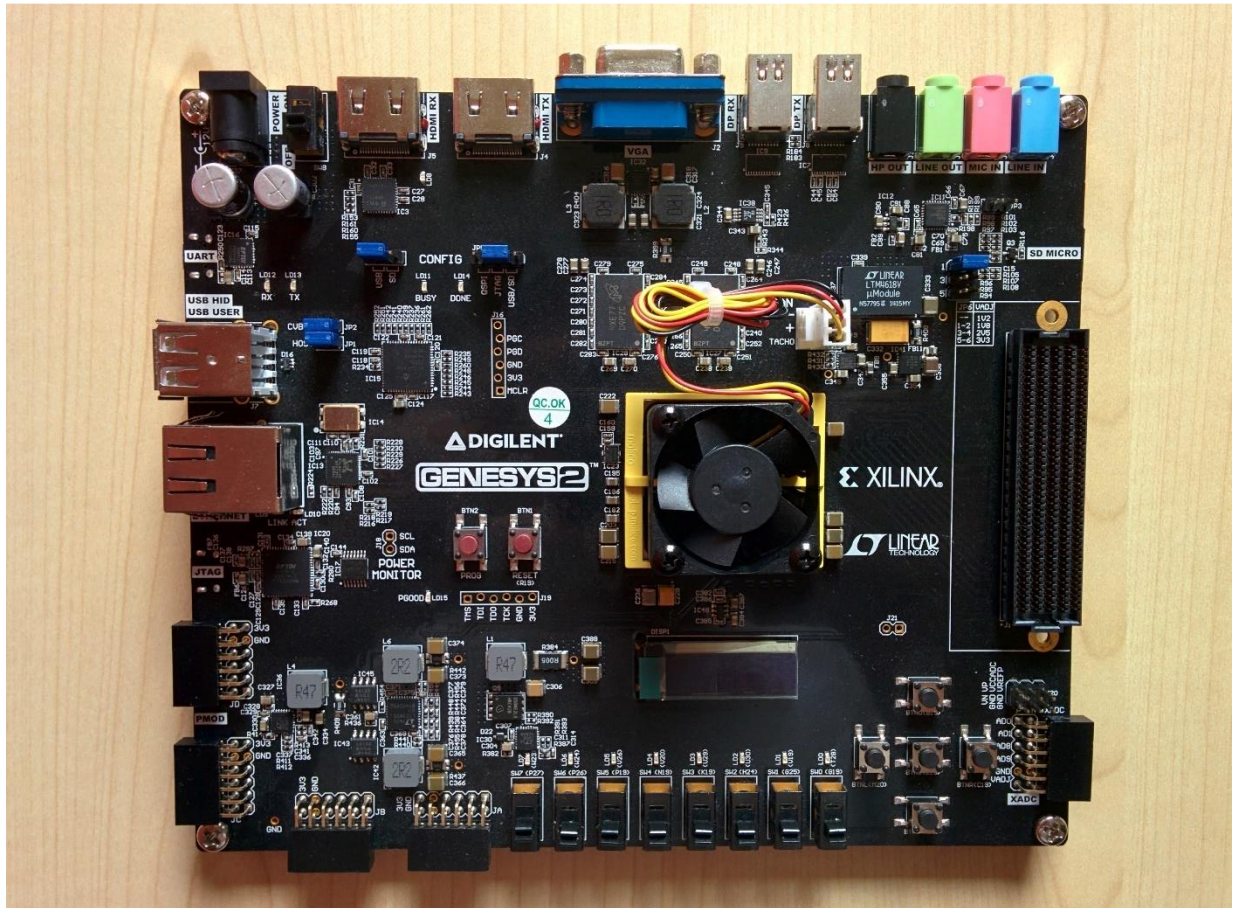
### 3.1 Kasutatavad tööriistad

#### 3.1.1 Genesys 2

Antud töös kasutatakse Digilent, Inc. toodetud Genesys 2 arendusplaati, mis kasutab Xilinx, Inc. seitsmenda seeria Kintex 7 XC7K325T-2FFG900C FPGA-d. Antud arendusplaat valiti, sest sellel on üks HDMI sisend ja üks väljund, et andmevoogu töödelda ja samal ajal edasi saata.



Lisaks on arendusplaadil mitmeid erinevaid võimalusi andmete salvestamiseks või saatmiseks. Näiteks on plaadil SD kaardi pesa, UART sild, Etherneti ja USB pesa ning ajutiseks salvestamiseks 1GB DDR3 SDRAM. Arendusplaat on disainitud just audio- ja videotöötlus-rakendusteks. [23]



Joonis 6. Genesys 2 arendusplaat

Arendusplaadil olev Kintex 7 XC7K325T on ehitatud 28 nm tehnoloogiaga. Sellel on üle 326 000 loogikaraku ja üle 50 000 loogikatüki. Iga tükk sisaldab nelja LUT-i ja kaheksat trigerit. Igal LUT-il on kuus sisendit ja kaks väljundit. Lisaks on kuni 4 000 Kb jagatud RAM-i ja kuni 16 000 Kb plokk-RAM-i. Antud FPGA-l on 500 sisend-väljundit, mis on jagatud kümne taktiregiooni vahel. Igas taktiregioonis on üks taktihaldus ala (CMT) taktsignaali genereerimiseks. [25]

### 3.1.2 VHDL ja Vivado

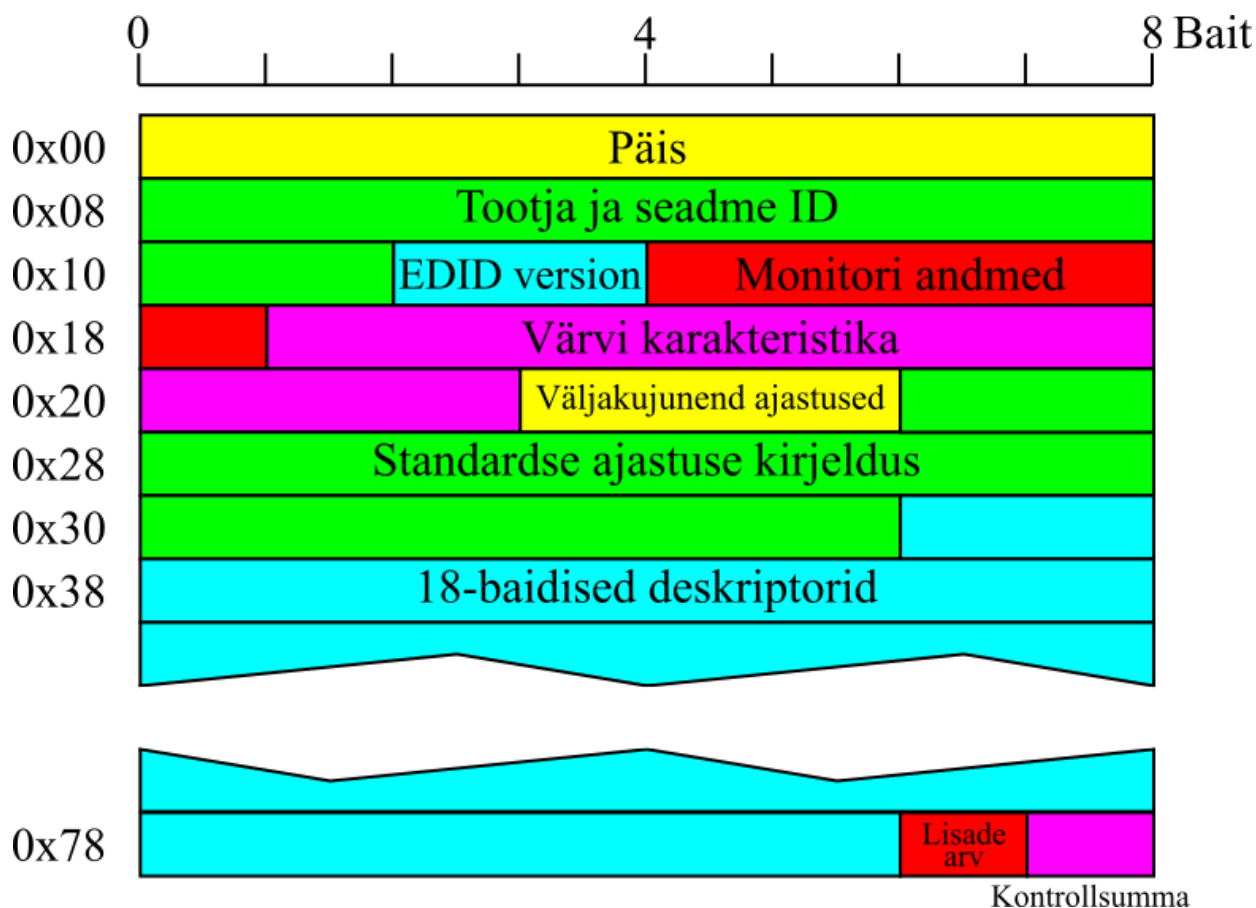
FPGA-de programmeerimiseks kasutatakse riistvarakirjelduskeelt. Põhilisteks kasutatavateks on VHDL ja Verilog, aga erinevaid keeli on palju. Antud töös kasutatakse VHDL-i, (VHSIC

*Hardware Description Language*), sest seda toetab kasutatav tööriist ja seda õpetatakse Tartu Ülikoolis. VHDL töötati välja Ameerika Ühendriikide Kaitseministeeriumi poolt 1980ndatel aastatel. Esimene versioon VHDL 87 anti välja 1988. aastal IEEE 1076 standardina [26]. Lisaks on olemas täiendav standard IEEE 1164 mis lisab mitme väärtusega loogikasüsteemide toe. [13]

Vivado on Xilinx, Inc. poolt väljaantav programm nende toodetud 7. seeria ja uuematele FPGA-dele, Zynq-7000 süsteemikiipidele ja teistele programmeeritavatele loogikaseadmetele tarkvara disainimiseks. Riistvarakirjelduskeeltest toetab Vivado VHDL-i ja Verilogi. Vivado pakub kõike vajalikku loogikaseadmetele tarkvara tegemiseks. Näiteks on Vivadosse sisse ehitatud riistvara simulaator kirjutatud koodi testimiseks. Veel on Vivados koodi sünteserimise, implementeerimise, bitijada genereerimise ja seadme programmeerimise funktsioon. Lisaks on võimalik muuta kirjutatava tarkvara täpsemaid seadeid, näiteks võimalust täpsustada viiteaegasid, planeerida programmeeritavate ühenduste radasid ja palju muud. [27]

## 3.2 EDID

EDID on 128-baidine infoplokk, mis sisaldab endas seadme andmeid. Üldjuhul on seadmeks monitor, aga käesolevas töös on selleks arendusplaat. EDID info saadetakse üle DDC radade I2C standardil, kasutades 100 kHz takti. EDID on vajalik, et saatja teaks, millist videosignaali saata ja kuidas seadet kasutajale kirjeldada. EDID algab 8-baidise päisega, seejärel on 10 baiti tootja ja seadme informatsiooni. Järgmisena on kahe baidiga ära mainitud EDID versiooni number. Videovoo baasinfo, nagu värvi tüüp, digitaalsus ja kuvasuhe, on kirjeldatud 5 baidiga. Värvi karakteristika on järgmises 10 baidis. Järgneva 50 baidiga määratakse vaikumisi resolutsioon ja teised võimalikud kasutatavad resolutsioonid. Lisaks on vajadusel võimalik kirjeldada seadme nime, seerianumbrit ja muud lisainformatsiooni. EDID saab ka laiendada täiendavate plokkidega. Nende arvu näitamiseks on eelviimane bait, millega piiratakse maksimaalne lisaplokkide arv 255-ni. Nii EDID enda kui ka lisaplokkide viimaseks baidiks on kontrollsumma. Selle väärtus arvutatakse nii, et kogu EDID 1-baidine summa peab olema 0. [9] [7]



Joonis 7. EDID struktuur [7]

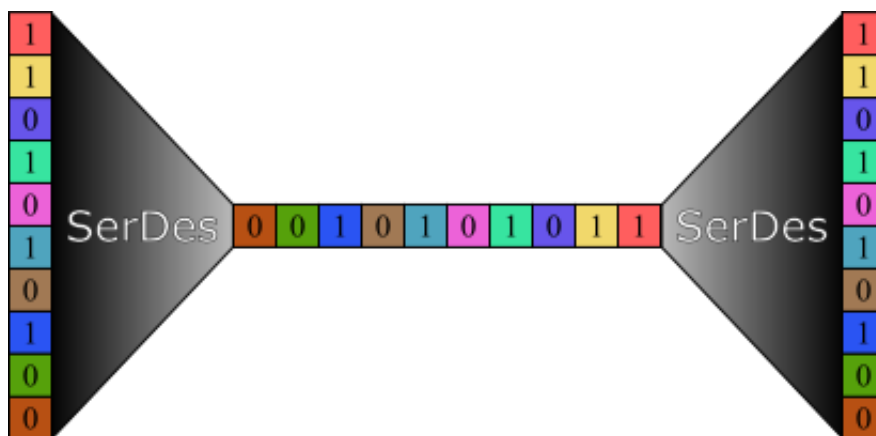
Antud töös kasutatav EDID on koostatud 1.4 spetsifikatsiooni järgi. Kokku pandud EDID tulemusena on arendusplaat kirjeldatud DVI seadmena, sest ei kasutata ühtegi lisaplokki. Tootjaks on Xilinx, Inc. ja monitori nimi on Gramann. Vaikimisi pildiresolutsioon on 1920 x 1080 ja kaadrisagedus 60Hz. Värvi formaat on RGB 4:4:4, milles iga piksli kohta saadetakse 24 bitti andmeid. Need 24 bitti on jagatud kolmeks nii, et iga RGB värvi kohta saadetakse 8 bitti infot. Kontrollsumma arvutamiseks kasutati programmi mis liitis kõik baidid 8 bitiste arvudena kokku ja lahutas saadud tulemuse nullist. EDID infot täiendades on võimalik seadme funktsioone laiendada juhaks, kui tekib vajadus näiteks teiste resolutsioonide järgi või audio salvestamiseks. Täielik EDID väärtuste tabel on saadaval lisas 2.

### 3.3 Andmevoo töötlus

#### 3.3.1 Paralleliseerimine

Andmeallikast tuleb videosignaali kolme bitijadana. Iga takti perioodiga tuleb iga jadaga 10 bitti. Bitijada paralleliseeritakse, sest andmeid töödeldakse 10-bitiste sümbolitena. Selleks otstarbeks

kasutatakse SerDes ehk jada-paralleelmuundurit (*Serializer/Deserializer*). SerDes saab sisendiks jadasignaali ja etalontakti ning väljastab paralleliseeritud andmed. Kasutatavas FPGA-s oleva ühe SerDesi väljund on kuni 8 bitti lai. Seega tuleb kasutada kahte SerDes ülem-alluv (*master/slave*) režiimis. Veel on vaja kasutada topelt kiirusega (DDR) režiimi, sest tavakiirusel (SDR – *Single Data Rate*) on andmelaius piiratud 8 bitini. See tähendab, et HDMI taktsignaali tuleb korrutada viiega, et seda kasutada paralleliseerimisel. Selleks kasutatakse MMCM-i HDMI takti taastamiseks ja korrutamiseks, millest pikemalt kirjas „Taktsignaali halduse“ peatükis. Antud töötlemise tulemusena saadakse kolm 10-bitist kodeeritud sümbolit. [9] [28]



Joonis 8. SerDesi põhimõte

### 3.3.2 Dekodeerimine

HDMI kasutab andmeedastuseks TMDS-i, mille eesmärk on vähendada biti väärtuste muutusi bitijadas, mis võimaldab tõsta andmeedastuskiirust. TMDS saadab infot 10-bitiste sümbolitena. Neist neli erinevat sümbolit vastavad kahe kontrollsignaali kõikidele võimalikele olekutele. Kui sisenev sümbol ei vasta ühele neist neljast sümbolist on teada, et sümbol vastab piksli värviinfole. 10-bitise sümboli kaks esimest bitti näitavad, kuidas signaali dekodeerida. TMDS-i dekodeerimise täpne loogika on näha lisas 3. [29]

Kõik sisenevad 10-bitised sümbolid ei ole aga TMDS sümbolid. Kui saabuv sümbol pole TMDS-i sümbol võib oletada, et sisenevad andmed ja taktsignaali pole sünkroniseeritud. Selliseid sümboleid aga lihtsa dekodeerijaga ei leia. Üks võimalus vigaste sümboleid leidmiseks on tabel, kus kõik võimalikud TMDS sümbolid on vastavuses 8-bitise sümboliga või kontrollsignaalidega. TMDS kasutab kokku 466 sümbolit. Ülejäänud võimaliku 558 sümbolit ei kasutata ja nende saabumisel võib oletada, et sisendandmed on valed. Tabeliga TMDS-i dekodeerimise puuduseks

on see, et kasutakse rohkem FPGA ressursse, aga samas on võimalik automaatselt saavutada sünkronisatsioon.

### 3.3.3 Sünkronisatsioon

FPGA-sse saabuval andmed ja takt võivad paralleliseerimise ajaks olla nihkesse läinud. Selle parandamiseks sisaldab iga Xilinx, Inc. 7. seeria FPGA sisend-väljund plokk programmeeritavat viiviselementi. Antud element annab võimaluse igat sisenevat signaali vajadusel viivitada. Viivise määramiseks on 5-bitine loendur, mis pakub viivise resolutsiooniks 32 erinevat väärtust. Viiviselementi kasutamine on vajalik koos SerDesiga. Lisaks on SerDesisse ehitatud bitinihke funktsionaalsus. Antud signaali sünkroonsel seadmisel nihkuvad väljundbitid ühe biti võrra. [28]

Antud töös kasutatakse Michael Alan Field kirjutatud loogikat sünkronisatsiooni saavutamiseks, mis reageerib vigaste TMDS sümbolite saabumisele. Kui saabub 12 vigast sümbolit umbes 12 miljoni sümboli jooksul, siis antud loogika suurendab 4-bitise loenduri väärtust. Kui toimub loenduri ületäitumine, siis sooritatakse SerDesi bitinihe ja nullitakse loendur. Kui on saabunud järjest suurem hulk korrektseid sümboleid, siis oletatakse et süsteem on sünkroonis. [30]

## 3.4 Andmete salvestamine

Kasutataval arendusplaadil on andmete salvestamiseks 1 GB DDR3 SDRAM, mis koosneb kahest 16 biti laiusest mälukomponendist, andes kokku 32-bitise andmesiini. Kasutatava mälu maksimaalne taktsagedus on 900 MHz ja maksimaalne andmevahetuskiirus umbes 1800 Mb/s. Antud mälu on piisavalt kiire piksli info jooksvalt salvestamiseks ja lugemiseks, aga selle kasutamise keerukus väljub käesoleva töö piiridest. [31]

Kui panna saadetav kaader kokku mitmest järjestikusest kaadrist, on võimalik kasutada ka FPGA-sse sisse ehitatud plokk-RAM, mille kogumaht on 16 000 Kb [25]. Selle puuduseks on aga asjaolu, et kiirelt liikuva pildi puhul tekib kaadrisse sisse nihe, mis on selgelt nähtav. Plokk-RAM on aga lihtsam ja ka majanduslikult soodsam kasutada, sest pole vaja välist mälukiipi. Üks plokkmälu kasutamise võimalus on kasutada seda FIFO-na. Ühe FIFO suurus on piiratud 36 Kb, aga neid on võimalik paralleelselt ja jadamisi kokku ühendada, mis kasvatab korraga kasutatavat mäluruumi. [28]

Kasutataval salvestamise loogikal on mitmeid funktsiooni. Esiteks loeb see jooksvalt saadetud piksli koordinaati, et vajadusel kaadril salvestamist pooleli jäänud asukohast jätkata. Teine funktsioon salvestab siseneva piksli andmed kolme FIFO-sse, üks FIFO ühe RGB värvi kohta.

Veel jälgitakse FIFO-de täituvust, et andmeid kaduma ei läheks. Kui FIFO-d on täis, salvestatakse viimasena salvestatud piksli koordinaat ja jäädakse ootama järgmist kaadrit. Viimaseks funktsiooniks on salvestatud andmete õiges järjekorras saatmine FIFO-dest DPTI kontrollerrisse. Lisaks jälgitakse, et DPTI kontrollerris olev FIFO ei täituks üle.

### 3.5 Andmevahetus

Arendusplaat ja arvuti suhtlevad omavahel DPTI (*Digilent Parallel Transfer Interface*) vahendusel. DPTI on Digilent, Inc. poolt välja töötatud liides andmeedastuseks nende toodetavatele arendusplaatidele. Kasutataval arendusplaadil on liideseks FTDI2232 USB kontrollerr, mis kasutab 8-bitist kahesuunalist andmesiini ja viit kontrollsignaali, et tekitada sünkroniseeritud paralleelühendus arvuti ja arendusplaadi vahele. USB kontrollerr väljutab 60 MHz taktsignaali andmete kirjutamiseks ja lugemiseks ning andmerada on 8 biti lai. Selle tulemusel on DPTI maksimaalne kiirus 480 Mb/s, mis on ka USB 2.0 maksimum. Ainuke muudatus, mis tuleb Digilent, Inc. koostatud koodis teha, on muuta kasutatav MMCM PLL-iks, sest taktsignaali sisendid on arendusplaadil samas sisend-väljund plokis, kus kasutamiseks on ainult üks CMT, mis sisaldab ühte MMCM-i ja ühte PLL-i. [6]

Arvuti poolel on Digilent Adept SDK programm, mis kirjutab ja loeb andmeid üle USB. See SDK leiab seadme aadressi, loob ühenduse, ning kirjutab ja loeb seadmest infot. [32] SDK-d kasutatav arvutiprogramm saadab seadmele käsu pildi salvestamise ja saatmise alustamiseks. Kasutatav andmeedastuskiirus ei võimalda kadudeta videovoo pidevat saatmist, mille tõttu on väljundiks üks kaader.

Arvutiprogrammi väljundiks on BMP (*Bitmap*) pildifail. BMP on rastergraafika failitüüp, kus on kirjas iga piksli väärtus. Seda on pärast TMDS dekodeerimist lihtne kasutada, sest väljundid on samad – iga piksli kohta 24 bitti RGB andmeid. Salvestades on piksliväärtuste järjekord sinine, roheline ja lõpuks punane. BMP fail koosneb päisest, kus on kogu pildi kohta käiv info ja kolm baiti iga piksli kohta, näiteks pildi resolutsioon, millist värviformaati kasutatakse, mis on kogu faili suurus ja palju muud. [3]

Antud töös kasutatava vormingu päise pikkus on 53 baiti ja määratletud pildi resolutsiooniks 1920 x 1080. Lõplik päis on saadaval lisas 4. Kasutatava BMP vormingu puudus on see, et pilti ei tihendata, mille tulemusel on failid suured. Käesoleva töö väljundiks oleva pildi suurus on näiteks 6 220 854 baiti.

## 3.6 Taktsignaaside haldus

Kirjutatud loogika tööks on vaja kasutada palju erinevaid taktsignaale. Peamiseks taktiks on 200 MHz, mille tarbeks on arendusplaadil ostsillaator, mille väljundiks on diferentsiaalne signaal. Seda takti kasutatakse andmete saatmisel DPTI kontrollerisse. Veel tehakse sellest 100 MHz takt EDID andmete saatmise loogika tarbeks. HDMI andmetega töötlemiseks kasutatakse piksli taktsignaali, mida saadetakse üle HDMI kaabli. HDMI takt viiekordistatakse andmete paralleliseerimiseks. USB kontroller saadab FPGA-sse 60 MHz taktsignaali andmete saatmiseks. Kõik antud seadmes kasutatavad taktsignaalid on järgmised:

- 60 MHz – USB taktsignaal
- 100 MHz – EDID saatmise loogika
- 148,5 MHz – Piksli taktsignaal
- 200 MHz – Süsteemi taktsignaal
- 742,5 MHz – Paralleliseerimine

Xilinx, Inc. seitsmenda seeria FPGA-d on jagatud sisemiselt taktiregioonideks, mida ühel seadmel on kuni 24. Iga taktiregioon sisaldab sisend-väljund plokki 50 viiguga, ühte CMT-d (*Clock Management Tile*), loogikablokke ja horisontaalset taktirada (HROW), mis võimaldab juhtida 12 erinevat takti antud regioonis. Taktiregioone ühendab takti magistraal (*backbone*), mis jagab taktsignaale erinevate taktiregioonide vahel. [4]

### 3.6.1 Taktsignaaside marsruutimine

FPGA taktsignaaside marsruutimiseks on seitsmenda seeria Xilinx, Inc. seadmetes kasutada palju erinevaid tüüpe puhvreid, mida kasutatakse erinevates olukordades. FPGA-sse sisenevaid signaale on vaja puhverdada IBUF-idega. Ka IBUF-e on palju erinevaid. Osad on mõeldud andmesignaale ja teised taktsignaale. Lisaks on olemas diferentsiaalsete signaalide puhvrid, mis taastavad diferentsiaalse signaali. Andmesignaale puhvreid kasutatakse sisenevate või väljuvate signaalide puhul. [28]

Taktsignaaside puhvreid on signaalide puhvritest rohkem. Peamine puhver on BUFG, mis suunab takti magistraalile. Taktiregiooni sees kasutatakse BUFH-d, mis suunab taktsignaali HROW-sse. BUFR ja BUFMR on taktiregioonide puhvrid. Esimene neist teeb taktsignaali kasutatavaks ainult ühes taktiregioonis ja teine ka naabertaktiregioonis. On olemas ka BUFIO-d, mis suunavad taktsignaali sisend-väljund plokki, et seda kasutada näiteks SerDesi tööks. [4]

Üldjuhul puhvleid ise määrama ei pea, vaid seda teeb kasutatav tööriist, käesolevas töös Vivado, automaatselt. Kohad, kus neid ise määrama peab, on näiteks taktsignaalide genereerimisel MMCM-iga ja SerDesit kasutades. Näiteks peab SerDesi kasutatav taktsignaali olema manuaalselt viidud läbi BUFIO. [4]

### 3.6.2 Taktsignaali genereerimine

Xilinx, Inc. seitsmenda seeria FPGA-des on taktsignaali haldamiseks CMT. CMT sisaldab MMCM-i ja PLL-i. MMCM-i (*mixed-mode clock manager*) ja PLL-i (*phase-locked loop*) kasutatakse erinevate taktsignaali genereerimiseks, müra vähendamiseks ja väliste taktide taastamiseks. Nende tööpõhimõtte on põhiliselt ühesugune, aga MMCM sisaldab rohkem funktsioone. Näiteks saab MMCM-iga ühe takti sagedust murdudega jagada, sellel on üks väljund rohkem ja lisaks on neljale väljundile juba sisse ehitatud inverter. [4]

Antud töös kasutatakse ühte PLL-i ja kahte MMCM-i. Ühte MMCM-i kasutatakse 200 MHz taktist 100 MHz tegemiseks ja teist HDMI takti taastamiseks ning viiega korrutamiseks. PLL-i kasutatakse DPTI taktsignaali taastamiseks ja nihkega takti genereerimiseks. DPTI kontrolleri kasutab PLL-i, sest nii HDMI kui ka DPTI taktsignaali sisendid asuvad arendusplaadil samas sisend-väljund plokis, mis tähendab, et neil on kahepeale kasutada ainult üks CMT.

## 3.7 Tööga kaasas olevad failid

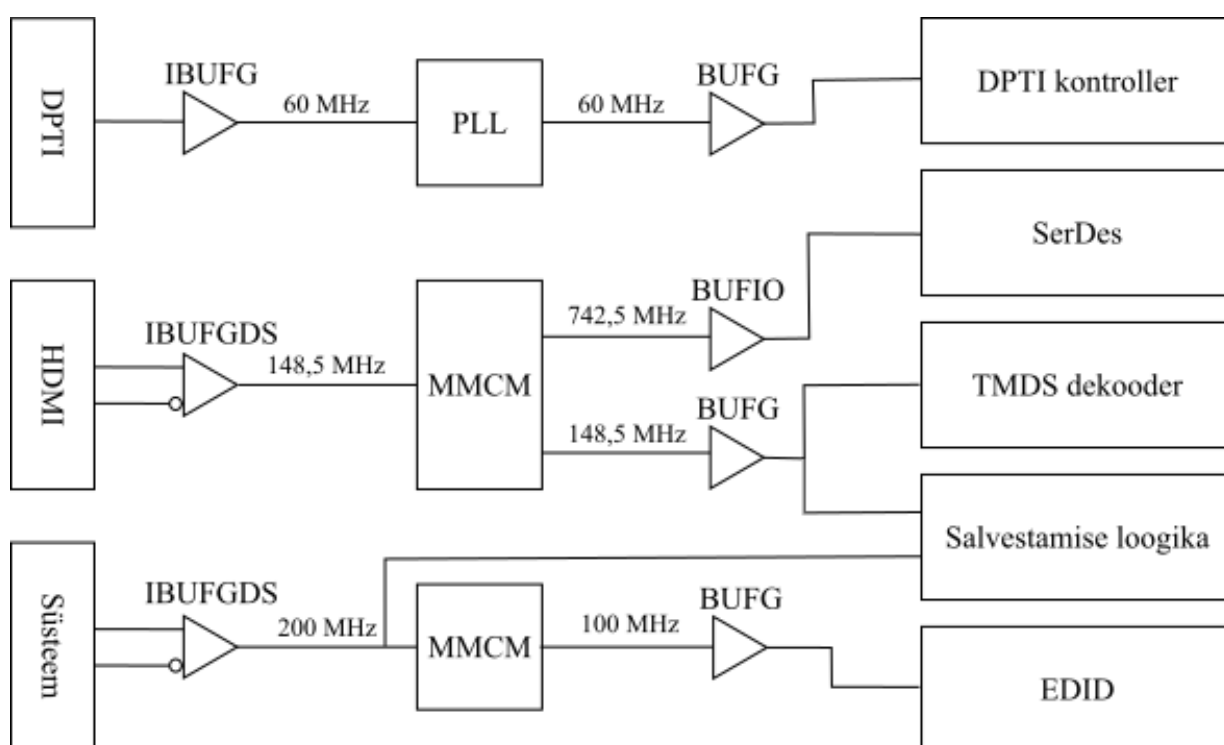
Käesoleva tööga on kaasas tihendatud kaust, mis sisaldab töö jätkamiseks vajalikke faile. Tihendatud failis HDMI\_capture on arhiveeritud Vivado projekt. Selle kasutamiseks tuleb see lahti pakkida ja sealt käivitada .xpr fail. Lähtefailid asuvad projekti HDMI\_capture.srcs kaustas. Kaustas DPTI on C++ kood arendusplaadi andmevahetuseks, mille kompileerimiseks on vaja Digilent Adept SDK-d. Kasutas on ka kompileeritud programm, mille käivitamisel küsitakse arendusplaadilt ühe 1920 x 1080 piksli suuruse kaadri jagu andmeid ja .bin fail, mis sisaldab endas BMP faili päist, nagu käesolevas töös kokku pandi. Programmi kasutamiseks peab olema arvutisse installitud Digilent Adept ja Microsoft Visual Studio. Programmi käivitamisel ja edukal andmevahetusel tekib kausta BMP fail salvestatud kaadriga. Kaust EDID sisaldab C-s kirjutatud koodi EDID kontrollsumma arvutamiseks.



## 4 Tulemused ja järeldused

### 4.1 Taktsignaalid

Jooniselt 9 on näha illustratiivne taktsignaalide võrgustik. Võrgustikus on peal kõik käsitsi määratud puhvrid, mida Vivado ise ei määra. Näha on taktsignaalide sisendpuhvrid, millest kaks on diferentsiaalsed sisendid, ning pärast taktsignaalide genereerimist kasutatavad puhvrid. Joonis näitab, et ainult SerDesi taktsignaali puhver on BUFIO. Ülejäänud puhvrid on BUFG-d.



Joonis 9. Taktsignaalide võrk

### 4.2 Kasutatud ressursid

Disainitud loogikaplokid kasutavad arendusplaadi FPGA-s olevaid ressursse väga vähe. LUT-idest ja trigeritest kasutatakse vähem kui 1%. Saadavalolevast plokki-RAM-ist kasutatakse 1,35%. Magistraalile suunatud takte on seitse, mis teeb võimalikust maksimumist 22%. Taktsignaalide genereerimiseks kasutatakse ühte PLL-i ja kahte MMCM-i. Mõlemaid on kokku 10. See tähendab, et kasutusprotsent on vastavalt 10 ja 20%. Kasutatud ressursside täpsed väärtused on nähtavad tabelis 1.

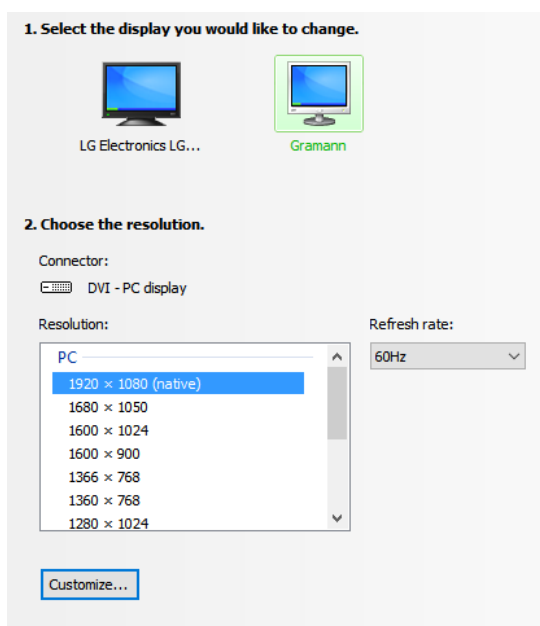
Ressurss	Kasutamine	Saadaval	Kasutusprotsent (%)
LUT	445	203800	0.22
FF	366	407600	0.09
BRAM	6	445	1.35
IO	47	500	9.40
BUFG	7	32	21.88
MMCM	2	10	20.00
PLL	1	10	10.00

*Tabel 1. FPGA ressursside kasutamine*

Madal FPGA ressursside kasutus tähendab kahte asja. Esiteks on loogikaelemente, et lisada täiendavaid funktsioone videohõiveks. Teiseks tähendab see, et kasutatava FPGA võib välja vahetada väiksema ja odavama vastu. Odavama FPGA valikul peab aga arvestama taktsignaalide nõuetega. Peamiseks takistuseks on paralleliseerimiseks kasutatav taktsignaal, mis 1920 x 1080 resolutsioonil ja 60 Hz kaadrisagedusel on 742,5 MHz.

### 4.3 Seadme tuvastamine

Kui programmeeritud arendusplaat ühendada HDMI kaabliga arvutiga, siis tuntakse see koheselt ära. Resolutsioon seatakse 1920 x 1080, kaadrisageduseks 60 Hz ja nimena kuvatakse „Gramann“. Lisaks on ühendusena määratud DVI, kuigi kasutatakse HDMI-d. Xilinx, Inc. monitori nime ees pole, sest tõenäoliselt puudub see levinud väärtuste tabelis. Väljavõtet arvuti videokaardi juhtpaneelilt, kui arendusplaat on ühendatud näeb jooniselt 10.



*Joonis 10. Arendusplaadi tuvastamine arvutis*

## 4.4 Kaadrihõive

Kui käivitada arvutis olev DPTI-ga suhtlev programm, siis alustatakse arendusplaadil kaadrihõivet. Jooksvalt salvestades ja saates jõuab üks kaader kohale umbes 7,7 sekundiga, nagu on näha jooniselt 11. See tähendab, et saadetud kaader on kokku pandud umbes 460 kaadrist. Saabunud kaadri infole lisatakse algusesse koostatud BMP päis ja salvestatakse pildina.

```
Synchronous DPTI Port Enabled
beginning data transfer...
writing to file
transferred 6220800 bytes in 7.688000 seconds, transfer rate = 809157.127992 B/sec, 790.192508
KB/sec, 0.771672 MB/sec
```

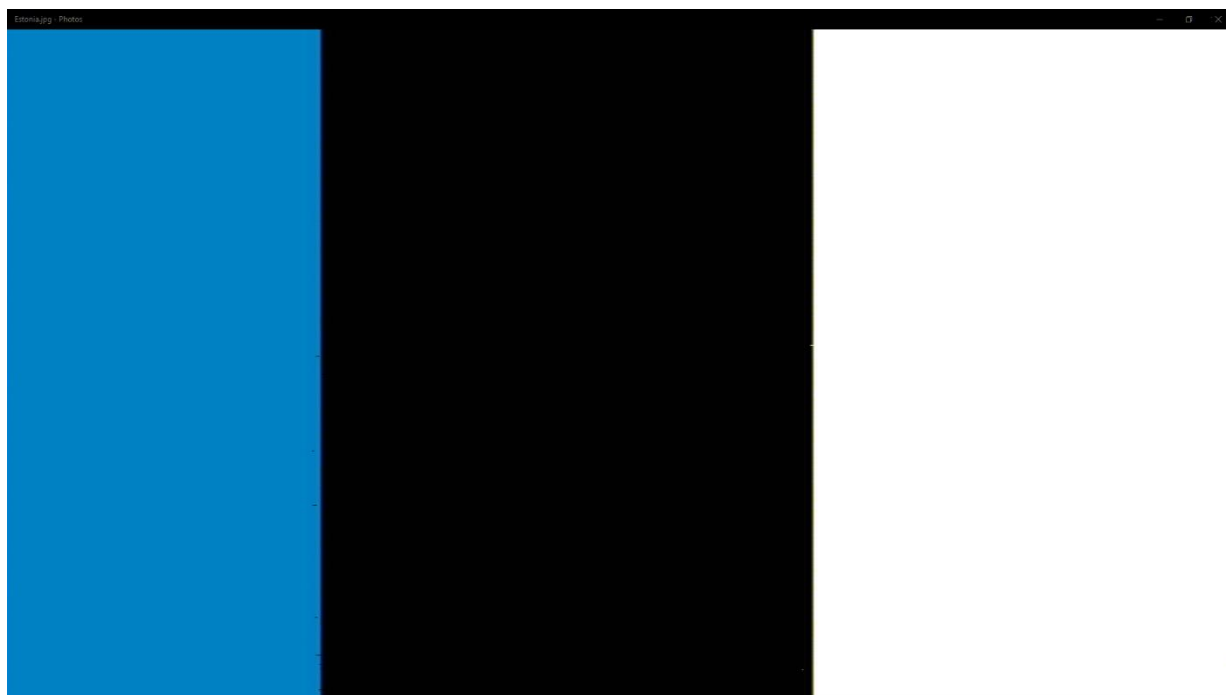
*Joonis 11. Väljavõte terminaliaknast pärast kaadri hõivamist*

Videoandmete saatjaks on Nvidia Corporation'i GeForce GTX 760 graafikakaart. Esimesena salvestatud kaadrite aluseks on vertikaalne Eesti lipp, mis on kuvatud 1920 x 1080 resolutsioonis. Joonisel 12 on esimene salvestatud kaader, millel on äratuntav kujutis. Salvestatud kaader nihkub vasakule, millest võib järeldada, et salvestamise või dekodeerimise loogikaga läheb informatsiooni kaduma. Selle pärast pole salvestatud piksliread piisavate pikkustega. Salvestatud värvid on korrektsed, mis tähendab, et salvestamise ja saatmise loogika vahel andmeid kaduma ei lähe.



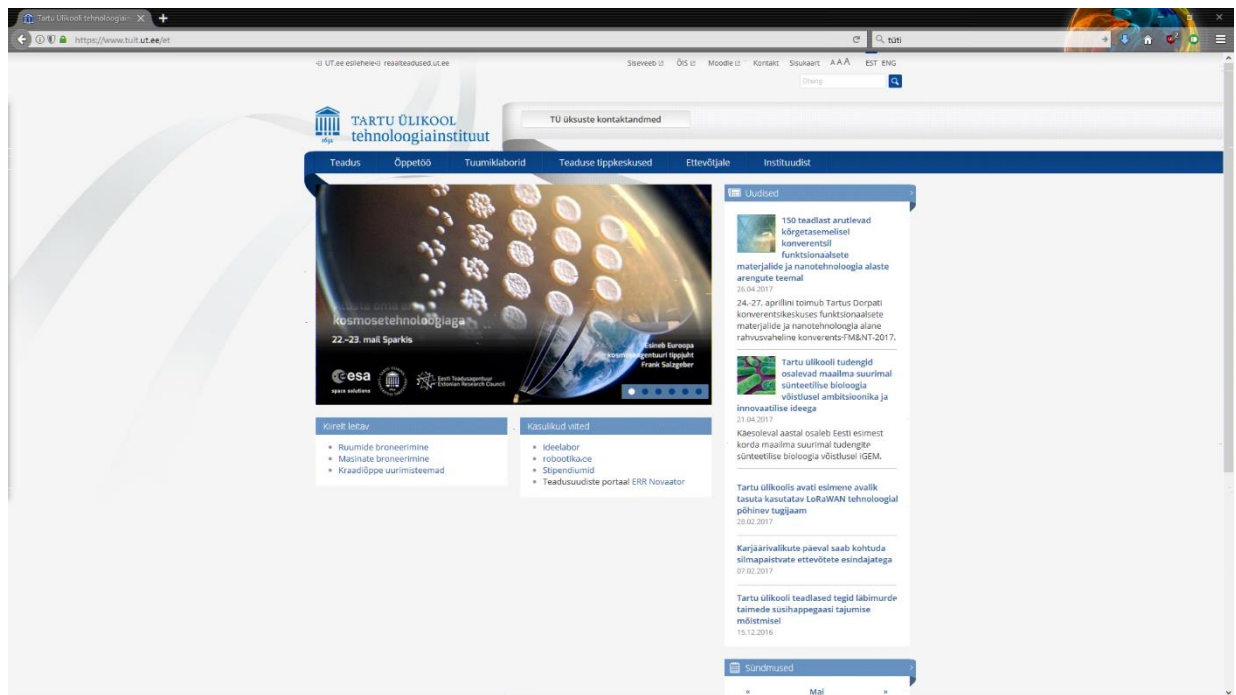
*Joonis 12. Esimene arusaadav kaader*

Pärast kasutatud dekodeerimise loogika korrastamist ja parandamist on esmasel vaatlemisel salvestatud kaader korrektne ja väiksemate pikslivigadeta. Ei ole märgata mingit andmete kadumist, mille tulemusel nihkuks salvestatud kaader vasakule ja ei salvestata ka liiga palju andmeid, mille tulemusel kalduks kaader paremale. Seega tekkinud nihke põhjus oli arvatavasti selles, et ei dekodeeritud õigesti video andmete perioodi piire. Pilti salvestatud kaadrist on näha joonisel 13.



*Joonis 13. Hõivatud kaader eesti lipust*

Väiksemate vigade nägemiseks tuleb salvestada detailsem kaader. Joonisel 13 on näha salvestatud kaadrit Tartu Ülikooli Tehnoloogiainstituudi kodulehest. Esiteks on näha veebilehel kuvatud pildi muutumist, sest see vahetub iga paari sekundi järel. See juhtub, sest kaader salvestatakse pikema aja jooksul, mille jooksul jõuab kuvatav pilt muutuda. Teiseks on peaaegu kogu veebilehel olev tekst loetav. Samas on osade tähtede ja värvimuutuste juures näha vigu. Veebilehe ühevärvilise tausta tõttu pole selliseid vigu palju märgata.



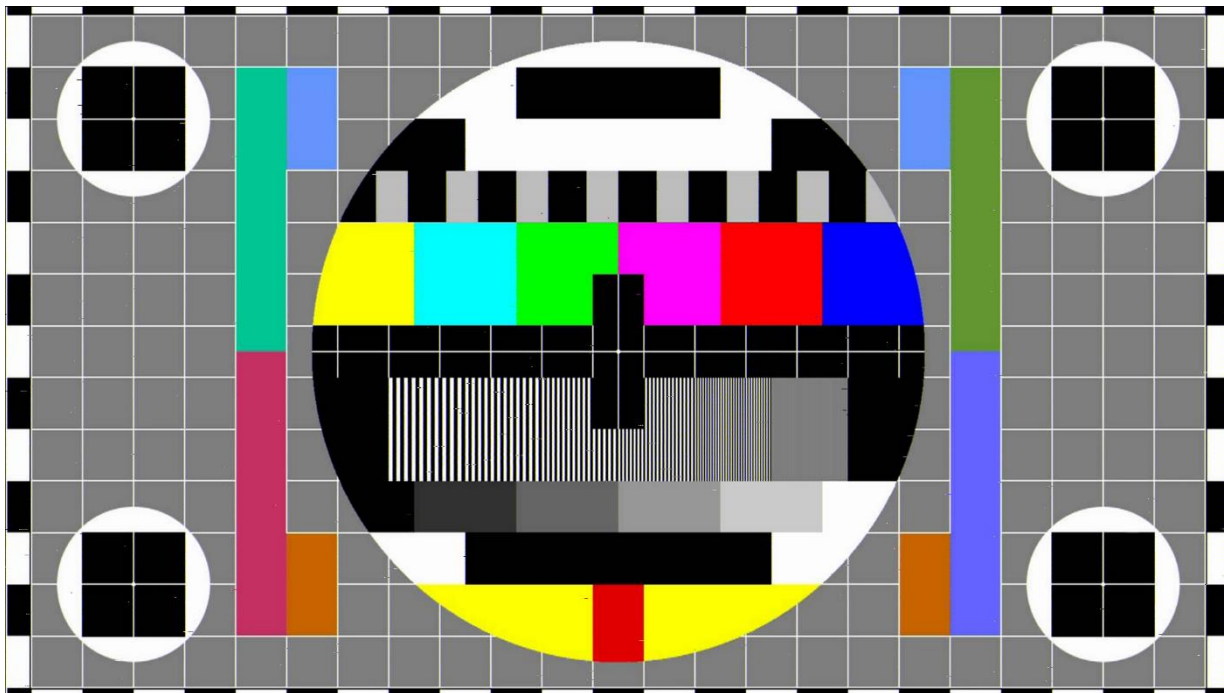
Joonis 14. Hõivatud kaader Tehnoloogiainstituudi kodulehest



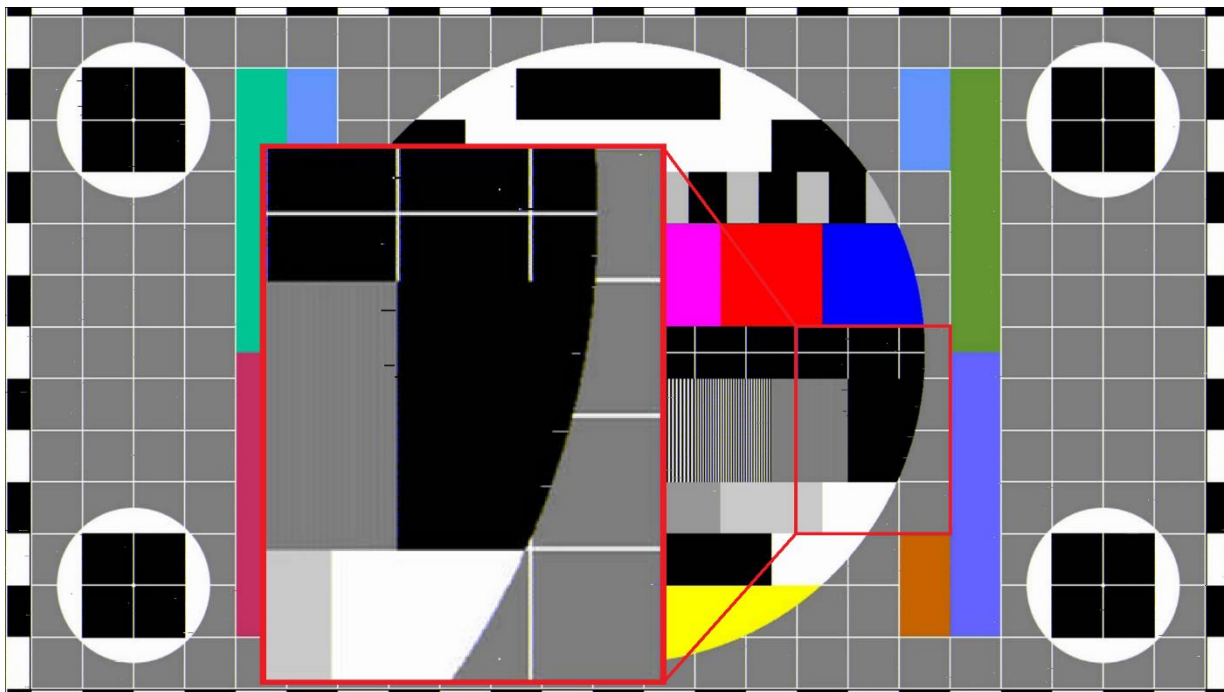
Joonis 15. Algne testmuster [33]

Vigade paremaks märgamiseks kasutatakse testmusterit, mida on näha joonisel 15. Joonisel 16 on arendusplaati kasutades salvestatud testmuster. Kiirel võrdlusel näivad pildid samasugused, aga lähemal vaatlemisel on näha, et salvestatud kaadril on väikesed vead sees. Ühte näidet sellest on suurendatult näha jooniselt 17. Suurendusel näeb, et kohati läheb informatsiooni kaduma, sest

vahtuvat värvi hakatakse salvestama liiga vara. Samas parandatakse viga kiiresti, sest üldpilt on terviklik. Võib oletada, et viga on salvestamisloogikas.



*Joonis 16. Arendusplaadiga salvestatud testmuster*



*Joonis 17. Hõivatud testmusteri vead suurendatult*

## 5 Kokkuvõte

Antud töö eesmärgiks oli teha eeltööd videohõiveks Genesys 2 arendusplaadil. Lisaks sooviti teostada digitaalse video vastuvõtja kasutades HDMI liidest, selle abil salvestada ühe kaadri jagu andmeid ning saata need üle USB arvutisse. HDMI liidest kasutades realiseeriti DVI ühendus digitaalsete videoandmete saatmiseks arvutist arendusplaadile. Veel realiseeriti EDID andmed seadme tuvastamiseks kuvarina ja TMDS dekodeer sisenevate andmete töötlemiseks. Nende komponentidega suudeti luua süsteem, millega saab hõivata ühe kaadri jagu infot mitme kaadri jooksul ja saata see üle USB arvutisse. Kaadri andmete ajutiseks salvestamiseks kasutati FPGA-sse ehitatud plokki-RAM-i.

Praktilises osas realiseeriti arendusplaadil EDID plokki arendusplaadi tuvastamiseks, SerDes sisenevate videoandmete paralleliseerimiseks ja TMDS dekodeer. Koostatud EDID ploki tulemusel saatis arvuti 1920 x 1080 resolutsioonil 60 Hz kaadrisagedusega pilti. Lisaks saavutati andmevahetus arendusplaadi ja arvuti vahel kasutades DPTI ühendust. Andmevahetuse abil saadeti ühe kaadri jagu infot arendusplaadilt arvutisse, kus see salvestati. Kaadri salvestamise tarbeks koostati BMP faili päis, et kokku panna 1920 x 1080 suurune BMP pildifail.

Projekti edasiarendamiseks on vaja läbida palju erinevaid etappe. Esiteks tuleks teostada välise RAM kasutamine, et kogu kaader korraga salvestada. Selleks peab implementeerima piisavalt kiire mäluliidese, mis ei tohi olla aeglasem kui sisenev videovoog. Teiseks tuleks implementeerida videotihendusalgorithm. See aitaks vähendada vajalikku andmeedastuskiirust arendusplaadi ja arvuti vahel. Kolmandaks tuleks lisada erinevate resolutsioonide ja kaadrisageduste tugi. Selleks peab taktigeneraator olema jooksvalt seadistatav. Lisaks võiks implementeerida audiosalvestamise võimaluse koos samaaegselt saadetava videovooga. Selle saavutamiseks tuleb täiendada EDID, et seade hakkaks audiot saatma üle HDMI. Lisaks peab muutma TMDS dekodeerijat, et see suudaks dekodeerida audioandmeid.

# Kasutatud kirjandus

- [1] A. Michael, „The Ultimate Guide to 4K and 8K Ultra HD,“ Ultra HDTV, 21 Mai 2015. [Võrgumaterjal]. Saadaval: <http://www.ultrahd.tv/articles/the-ultimate-guide-to-4k-and-8k-ultra-hd/>. [Kasutatud 4 Mai 2017].
- [2] M. Rouse, „ASIC (application-specific integrated circuit),“ TechTarget, September 2005. [Võrgumaterjal]. Saadaval: <http://whatis.techtarget.com/definition/ASIC-application-specific-integrated-circuit>. [Kasutatud 6 Mai 2017].
- [3] DrAgon, „Simplified Windows BMP Bitmap File Format Specification,“ DragonWins, 27 November 2010. [Võrgumaterjal]. Saadaval: <http://www.dragonwins.com/domains/getteched/bmp/bmpfileformat.htm>. [Kasutatud 13 Aprill 2017].
- [4] Xilinx, Inc., 7 Series FPGAs Clocking Resources User Guide (UG472), Xilinx, Inc., 2017.
- [5] Digital Display Working Group, Digital Visual Interface DVI, 1.0 ed., 1999.
- [6] Digilent, Inc., Digilent Parallel Transfer Interface (DPTI), Pullman: Digilent, Inc., 2016.
- [7] Video Electronics Standards Association, VESA Enhanced Extended Display Identification Data Standard, Milpitas: VESA, 2006.
- [8] Xilinx, Inc., „Field Programmable Gate Array,“ Xilinx, Inc., 11 Juuli 2012. [Võrgumaterjal]. Saadaval: <https://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htm>. [Kasutatud 15 Märts 2017].
- [9] Hitachi, Matsushita Electric Industrial Co., Philips Consumer Electronics, International B.V., Silicon Image, Sony Corporation, Thomson Inc., Toshiba Corporation, High-Definition Multimedia Interface, 2006.



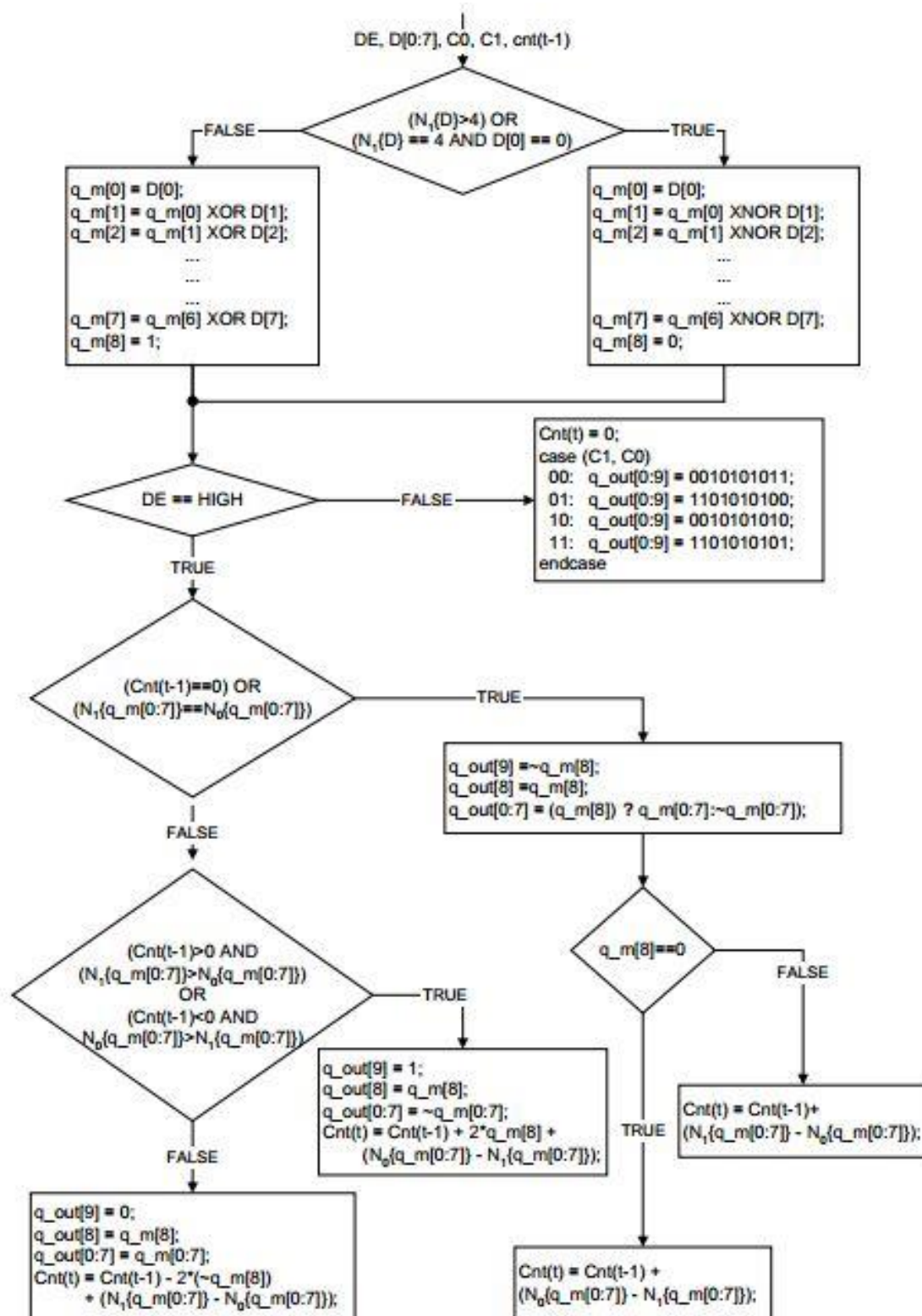
- [10] I2C Bus, „I2C – What’s That?“, I2C Bus, [Võrgumaterjal]. Saadaval: <http://www.i2c-bus.org/i2c-bus/>. [Kasutatud 4 Mai 2017].
- [11] M. Rouse, „Phase-Locked Loop“, TechTarget, September 2016. [Võrgumaterjal]. Saadaval: <http://searchnetworking.techtarget.com/definition/phase-locked-loop>. [Kasutatud 6 Mai 2017].
- [12] A. Patel, „The basics of SerDes (serializers/deserializers) for interfacing“, Planet Analog, 16 September 2010. [Võrgumaterjal]. Saadaval: [http://www.planetanalog.com/document.asp?doc\\_id=528099](http://www.planetanalog.com/document.asp?doc_id=528099). [Kasutatud 6 Mai 2017].
- [13] V. A. Pedroni, „About VHDL“, *Circuit Design with VHDL*, Cambridge, Massachusetts, MIT Press, 2004, p. 3.
- [14] A. Walgrove, „The Explosive Growth of Online Video, in 5 Charts“, Contently, 6 Juuli 2015. [Võrgumaterjal]. Saadaval: <https://contently.com/strategist/2015/07/06/the-explosive-growth-of-online-video-in-5-charts/>. [Kasutatud 27 Aprill 2017].
- [15] Alexa, „The top 500 sites on the web“, 27 Aprill 2017. [Võrgumaterjal]. Saadaval: <http://www.alexa.com/topsites>. [Kasutatud 27 Aprill 2017].
- [16] D. O. Hodgson ja D. B. Gochner, „Video capture expansion card“. US Patent 5550566, 27 08 1996.
- [17] Magwell, „Magewell Adopts Xilinx FPGA for Pro Capture Family“, Magwell, 21 Aprill 2016. [Online]. Saadaval: <http://www.magewell.com/3359.html>. [Accessed 19 Veebruar 2017].
- [18] Intel FPGA, „Acquisition - Capture Cards“, Intel Corporation, 11 Detsember 2016. [Võrgumaterjal]. Saadaval: <https://www.altera.com/solutions/industry/broadcast/applications/acquisition/capture-cards.html>. [Kasutatud 9 Märts 2017].
- [19] HDMI Forum, „Introducing HDMI 2.1 Specification“, HDMI Forum, 2017. [Võrgumaterjal]. Saadaval: [http://www.hdmi.org/manufacture/hdmi\\_2\\_1/index.aspx](http://www.hdmi.org/manufacture/hdmi_2_1/index.aspx). [Kasutatud 24 Veebruar 2017].

- [20] Texas Instruments, TMDS141 HDMI Hider, 2011.
- [21] N. Eastland, „FPGA – Configurable Logic Block,“ Digilent Blog, 6 August 2015. [Võrgumaterjal]. Saadaval: <https://blog.digilentinc.com/fpga-configurable-logic-block/>. [Kasutatud 15 Märts 2017].
- [22] „FPGAs – Internal RAM,“ fpga4fun, [Võrgumaterjal]. Saadaval: <http://www.fpga4fun.com/FPGAinfo3.html>. [Kasutatud 15 Märts 2017].
- [23] Digilent, Inc., „Genesys 2 Kintex-7 FPGA Development Board,“ Digilent, Inc., [Võrgumaterjal]. Saadaval: <http://store.digilentinc.com/genesys-2-kintex-7-fpga-development-board/>. [Kasutatud 7 Märts 2017].
- [24] Xilinx, Inc., „FPGA vs. ASIC,“ Xilinx, Inc., [Võrgumaterjal]. Saadaval: <https://www.xilinx.com/fpga/asic.htm>. [Kasutatud 10 Mai 2017].
- [25] Xilinx, Inc., 7 Series FPGAs Data Sheet: Overview (DS180), Xilinx, Inc., 2016.
- [26] IEEE, 1076-1987 - IEEE Standard VHDL Language Reference Manual, IEEE Xplore®, 1988.
- [27] Xilinx, Inc., Vivado Design Suite User Guide: Using the Vivado IDE (UG893), Xilinx, Inc., 2016.
- [28] Xilinx, Inc., Xilinx 7 Series FPGA and Zynq Libraries Guide for HDL Designs (UG768), Xilinx, Inc., 2012.
- [29] Silicon Image, Digital Visual Interface & TMDS Extensions, 2004.
- [30] M. A. Field, „Artix 7 HDMI processing,“ GitHub, 22 Juuli 2016. [Võrgumaterjal]. Saadaval: <https://github.com/hamsternz/Artix-7-HDMI-processing>. [Kasutatud 30 Märts 2017].
- [31] Digilent, Inc., „Genesys 2 Reference Manual,“ Digilent, Inc., 20 Märts 2017. [Võrgumaterjal]. Saadaval: <https://reference.digilentinc.com/reference/programmable-logic/genesys-2/reference-manual>. [Kasutatud 16 Aprill 2017].

- [32] Digilent, Inc., Digilent Adept Parallel Interface (DPTI) Programmer's Reference Manual, Pullman: Digilent, Inc., 2015.
- [33] RTFMASAP, Artist, *PM5644-1920x1080*. [Art]. Wikimedia Commons, 2008.
- [34] Unified Extensible Firmware Interface Forum, „PNP ID Registry,“ UEFI Forum, 27 Aprill 2016. [Võrgumaterjal]. Saadaval: [http://www.uefi.org/pnp\\_id\\_list](http://www.uefi.org/pnp_id_list). [Kasutatud 10 Aprill 2017].
- [35] B. O'Donnel, White Paper - HDMI™ : The Digital Display Link, IDC, 2006.
- [36] S. Gábor, „About FPGAs,“ 13 Detsember 1999. [Võrgumaterjal]. Saadaval: <http://home.mit.bme.hu/~szedo/FPGA/fpgahw.htm>. [Kasutatud 15 Märts 2017].

# Lisad

## Lisa 1 TMD5 kodeerija loogika [5]

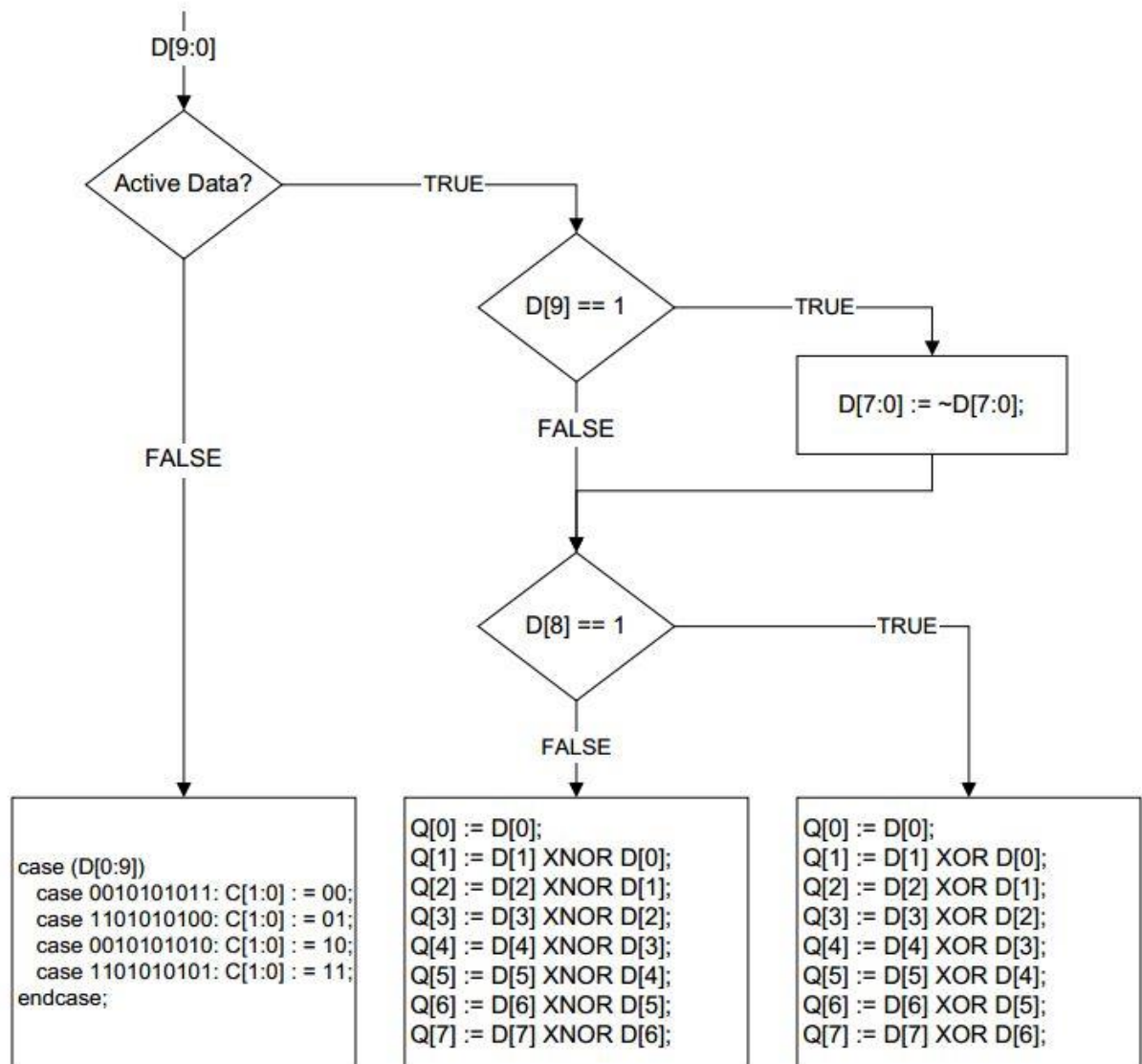


## Lisa 2 EDID väärtuste tabel

Aadress	Kirjeldus	Väärtus
<b>0x00-0x07</b>	<b>Päis</b>	0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00
<b>0x08-0x11</b>	<b>Tootja ja seadme ID</b>	
0x08-0x09	Tootjakood	0x61, 0x98 („XLX“ – Xilinx, Inc.) [34]
0x0A-0x0B	Tootekood	0x34, 0x12 (0x1234)
0x0C-0x0F	Seerianumber	0x01, 0x00, 0x00, 0x00 (0001)
0x10	Tootmise nädal	0x01 (1)
0x11	Tootmise aasta	0x1B (2017 – Loetakse alates 1990)
<b>0x12-0x13</b>	<b>EDID versioon</b>	0x01, 0x04 (1.4)
<b>0x14-0x18</b>	<b>Monitori andmed</b>	
0x14	Videosisendi info	0xA2 (Digitaalne, 8bit, HDMI-a)
0x15-0x16	Kuvasuhe	0x4F, 0x00 (16:9)
0x17	Gamma	0x78 (2.2)
0x18	Lisavõimalused	0x26 (RGB 4:4:4, sRGB)
<b>0x19-0x22</b>	<b>Värvi karakteristik</b>	0xEE, 0x91, 0xA3, 0x54, 0x4C, 0x99, 0x26, 0x0F, 0x50, 0x54
<b>0x23-0x25</b>	<b>Väljakujunenud ajastuste info</b>	0x20, 0x00, 0x00 (640×480 @ 60 Hz)
<b>0x26-0x35</b>	<b>Standardse ajastuse kirjeldus</b>	0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01 (Ei kasutata)
<b>0x36-0x6C</b>	<b>18-baidised deskriptorid</b>	
0x36-0x47	Eelistatud resolutsioon ja kaadrisagedus	0x02, 0x3A, 0x80, 0x18, 0x71, 0x38, 0x2D, 0x40, 0x58, 0x2C, 0x04, 0x05, 0x0F, 0x48, 0x42, 0x00, 0x00, 0x1E (148,5 Mhz takt, 1920X1080 @ 60 Hz)

0x48-0x59	Monitori nimi	0x00, 0x00, 0x00, 0xFC, 0x00, 0x47, 0x72, 0x61, 0x6D, 0x61, 0x6E, 0x6E, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20 (Gramann)
0x5A-0x6B	Lisa tekst	0x00, 0x00, 0x00, 0xFE, 0x00, 0x42, 0x61, 0x6B, 0x61, 0x6C, 0x61, 0x75, 0x72, 0x65, 0x75, 0x73, 0x0A, 0x20 (Bakalaureus)
0x6C-0x7D	Tühi väli	0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
<b>0x7E</b>	<b>Lisade arv</b>	0x00 (0)
<b>0x7F</b>	<b>Kontrollsumma</b>	0x72

## Lisa 3 TMDS dekodeeri loogika [5]



## Lisa 4 BMP faili päis

Nihe	Kirjeldus	Väärtus
<b>0x00</b>	<b>Faili päis</b>	
0x00	Faili tüüp	0x42, 0x4D („BM“ - <i>Bitmap</i> )
0x02	Faili suurus	0x36, 0xEC, 0x5E, 0x00 (6 220 854 baiti)
0x08	Kasutamata	0x00, 0x00, 0x00, 0x00
0x0A	Piksli maatriksi alguse	0x36, 0x00, 0x00, 0x00
<b>0x0E</b>	<b>Pildi päis</b>	
0x0E	Pildi päise suurus	0x28, 0x00, 0x00, 0x00 (40 baiti)
0x12	Pildi laius pikslites	0x80, 0x07, 0x00, 0x00 (1920)
0x16	Pildi kõrgus pikslites	0xC8, 0xFB, 0xFF, 0xFF (-1080 – negatiivne väärtus, et loetaks pilti ülevalt alla)
0x1A	Värvitasandite arv	0x01, 0x00 (alati 1)
0x1C	Bitte piksli kohta	0x18, 0x00 (24 biti)
0x1E	Tihendamise tüüp	0x00, 0x00, 0x00, 0x00 (0 – ilma tihendamiseta)
0x22	Pikslimaatriksi suurus	0x00, 0xEC, 0x5E, 0x00 (6 220 800 baiti)
0x26	Printimisresolutsioon	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 (Ei kasuta)
0x2E	Värvide arv värvitabelis	0x00, 0x00, 0x00, 0x00 (0 – ei kasuta värvitabelit)
0x32	Tähtsad värvid	0x00, 0x00, 0x00, 0x00 (0 – kõik värvid tähtsad)
<b>0x36</b>	<b>Pikslimaatriksi algus</b>	



# **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Jüri Gramann

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

## **„Eeltöö videohõiveks Genesys 2 arendusplaadil“**

mille juhendaja on Margus Rosin

- 1.1 reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2 üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
  3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 17.05.2017